PCT/AU00/01296

4

Patent Office
Canberra

I, JONNE YABSLEY, ACTING TEAM LEADER EXAMINATION SUPPORT & SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. PQ 3603 for a patent by ACTIVESKY, INC. filed on 22 October 1999.

WITNESS my hand this
Twentieth day of November 2000

J R Yabsley

JONNE YABSLEY
ACTING TEAM LEADER
EXAMINATION SUPPORT & SALES

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

P/00/009 28/5/91
Regulation 3.2

UTOVIA, INC.

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

for the invention entitled:

"An Object Oriented Video System"

The invention is described in the following statement:

AN OBJECT ORIENTED VIDEO SYSTEM

The present invention relates to a video encoding method, and in particular, but not exclusively, to a video encoding system which incorporates object orientated controls into encoded video streams that can be decoded by mobile computer devices, such as personal digital assistance (PDAs), hand held computers and custom wearable computing devices.

Recent technology improvements have resulted in the introduction of personal mobile computing devices, which are just beginning to include full wireless communication technologies. The global uptake of wireless mobile telephones has been significant, but is ready for substantial growth. There have, however not been any video technology solutions which have provided the video quality, frame rate or low power consumption required for potential new and innovative mobile video processes. There are currently no suitable mobile video solutions for processes utilising personal computing devices such as mobile video conferencing, ultra thin wireless network client computing, broadcast wireless mobile video, mobile video promotions and wireless video surveillance.

Computer based video conferencing currently uses standard computer workstations or PCs connected through a network consisting of a physical cable connection and network computer communication protocol layers. An example of this is a videoconference between two PCs over the Internet, with physically connected cables end to end, using the TCP/IP network communication protocols. This kind of video conferencing requires a physical connection to the Internet and also requires the use of large computer based video monitoring equipment. It provides for a videoconference between fixed locations, which additionally constrains the participants to a specific time for the conference to ensure that both parties will be at the appropriate locations simultaneously.

Broadcast of wireless digital textual information for personal hand held computers has only just become feasible with advances in new and innovative wireless technologies and hand held computing devices. Hand held computing devices and mobile telephones are able to have wireless connections to wide area

networks that can provide textual information to the user device. There is currently no real time broadcast of video to wireless hand held computing devices. One important market issue for broadcast media in any form is the question of advertising and how it is to be supported. Effective advertising should be specifically targeted to both users and locations.

Current video broadcast systems are unable to embed targeted advertising because of the enormous processing requirements needed to insert the advertising material into the video data streams in real time during transmission. The alternate method of precompositing video prior to transmission is too tedious to be performed on a regular basis. Additionally once the advertising is embedded into the video stream the user is unable to interact with the advertising in any manner, which reduces the effectiveness of the advertising. Significantly more effective advertising can be achieved though interactive means.

Commercial and domestic security based video surveillance systems have to date been achieved using closed circuit monitoring systems with video monitoring achieved in a central location requiring the full time attention of a dedicated surveillance guard. Video monitoring of multiple locations can only be achieved at the central control centre using dedicated monitoring system equipment. Security guards have no access to video from monitored locations, whilst on patrol.

Network based computing using thin client workstations involves minimal software processing on the client workstation, with the majority of software processing occurring on a server computer. Thin client computing reduces the cost of computer management due to the centralisation of information and operating software configuration. Client workstations are physically wired through standard local area networks such as 10 Base T Ethernet to the server computer. Client workstations run a minimal operating system enabling communication to the backend server computer and information display on the client video monitoring equipment. Existing systems however are constrained. They are typically limited to specific applications or vender software. For example current thin clients are unable to simultaneously service a video being displayed and a spreadsheet application.

To directly promote product in the market, sales representatives can use video demonstrations to illustrate product usage and benefits. Currently, for the mobile sales representative this involves the use of cumbersome, large, dedicated video display equipment, which can be taken to customer locations for product demonstrations. There are no mobile hand held video display solutions available, which provide real time video for product and market promotional purposes.

The present invention relates to a video encoding method, including:

quantising colour data in a video stream based on a reduced representation of colours;

generating motion vectors representing colour changes in a video frame of said stream;

generating encoded video frame data representing said quantised colours, motion vectors and transparent regions; and

generating encoded audio data and object control data for transmission with said encoded video data.

Advantageously said object control data represents parameters for rendering objects of a video frame. The parameters may represent object transparency, scale, volume, position and rotation. The control data may also represent parameters of a scene, such as size, number of objects and background colour.

Advantageously, the encoded video, audio and control data may be transmitted in respective packets for respective decoding.

The present invention also provides a video encoding method, including at least one of the following steps:

(i) selecting a reduced set of colours for each frame;

(ii) reconciling colours from frame to frame;

(iii) executing fast shape based motion compensation;

(iv) determining update areas of a frame based on a perceptual colour difference measure and layering;

(v) defining an update area using a hierarchal structure;

(vi) including further colour data to enhance image quality;

(vii) encoding a video data stream for each video object;

(viii) including in each video object animation and rendering controls for a user.

The present invention also provides a video decoding method for decoding the encoded data.

The present invention also provides a method of including targeted user and/or local video advertising.

The present invention also includes executing an ultrathin client, which may be wireless, and which is able to provide access to remote servers.

The present invention also provides a method for multivideo conferencing.

The present invention also provides systems for executing any one of the above methods, respectively.

The present invention also provides a computer program for executing the steps of any one of the above methods, respectively.

The present invention also provides a video on demand system. The present invention also provides a video security system. The present invention also provides an interactive mobile video system.

Preferably the above method and systems are based on one of the above video encoding methods.

The present invention also provides a computer program including code for controlling object orientated video and/or audio. Advantageously, the code may include IVML instructions.

Preferred embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, wherein:

Figure 1 is a block diagram of a preferred embodiment of an object-oriented video encoder.

Figure 2 is a block diagram of a preferred embodiment of an object-oriented video decoder.

Figure 3 is a block diagram of an input colour processing component of the video encoder

Figure 4 is a block diagram of perceptual colour reduction process component of the input colour processing component.

Figure 5 is a diagram of tree splitting used in the video encoder.

Figure 6 is a diagram of fast motion compensation used in the video encoder.

Figure 7 is a diagram of a region update selection process used in the video encoder.

Figure 8 is a diagram of pseudo random raster scanning used in the video encoder.

Figure 9 is a block diagram of an embodiment of an ultra-thin computing client Local Area wireless Network (LAN) system.

Figure 10 is a block diagram of an embodiment of an ultra-thin computing client Wide Area wireless Network (WAN) system.

Figure 11 is a block diagram of an embodiment of an ultra-thin computing client Remote LAN server system.

Figure 12 is a block diagram of an embodiment of an multiparty wireless videoconferencing system.

Figure 13 is a block diagram of an embodiment of an interactive video on demand system with targeted user video advertising.

Figure 14 is a block diagram of an embodiment of an interactive video on demand system with user authentication, access control, billing and usage metering.

Figure 15 is a block diagram of an embodiment of a video security/surveillance systems.

A video encoder, shown in Figure 1, executes an object-orientated video encoding process. The process is executed by eight main components of the encoder. The components can be implemented in software but to enhance the speed of the encoder , all the components are implemented in an application specific integrated circuit (ASIC) developed specifically to execute the steps of the encoding process. An input colour processing component 10 receives and processes individual input video frames and eliminates redundant and unwanted colours. An audio coding component 12 compresses input audio data using adaptive delta pulse code modulation (ADCPM) according to the ITU specification G.723. A scene/object control data component 14 encodes scene animation and presentation parameters associated with the input audio and video and which determine the relationships and behaviour of each input video object. A colour difference management and synchronisation component 16 receives the output of the input colour processor 10 and determines the encoding required using the previously encoded frame as a basis. The output is then provided to a combined spatial/temporal coder 18 to compress the video data. The output is provided to a decoder 20, which executes the inverse function to provide the frame to the colour management and synchronisation component 16 after a one frame delay 24. A transmission buffer 22 receives the output of decoder 18, the audio coder 12 and the control data component 14. The transmission buffer 22 manages transmission from a video server housing the encoder, by interleaving and

controlling data rates. If necessary, the encoded data can be encrypted by an encryption component 28 for transmission.

A video decoder, as shown in Figure 2, is able to receive and decode the data transmitted by the encoder of Figure 1. The decoder also includes a number of components to execute the decoding process. The steps of the decoding process are simplistic when compared to the encoding process and can be executed entirely by software compiled on a mobile computing device. An input data buffer 30 is used to hold the incoming transmitted data. The data is then forwarded to an input data switch 32, either directly or via a decryption unit 34, to determine the sub-processes 36 to 42 required to decode the data and then forward the data to the correct component that executes that sub-process. A colour management component 36 extracts colour information from the decoded data and separate components 38 and 42 perform video and audio decoding. An object management component 40 extracts object behaviour and appearance data for use in controlling the video scene. A video/object display component 44 renders visual objects on the basis of data received from the colour management, video decoding and object management components 36 to 40. An audio play back component 46 generates audio on the basis of data received from the audio decoding and object management component 40. A user input/control component 48 generates instructions and can control both the video and audio generated by the display and playback components 44 and 46. The user control component 48 is used to also transmit control messages back to the video server.

The input colour processing component 10 of the video encoder consists of two main parts, as shown in Figure 3. The first performs perceptual colour reduction and the other statistical colour reduction. The colour management component 16 manages the colour data generated by the first two processes of the component 10.

The statistical colour reduction may be implemented using various techniques including histogram (popularity), median cut, k-nearest neighbour and variance methods. The perceptual colour reduction makes use of psychophysical laws.

The perceptual colour reduction process is a nonlinear, perceptually uniform initial quantisation process that filters out the indistinguishable and visually redundant colours. This improves the performance of the statistical colour reduction process. To determine this initial quantisation the just noticeable difference (JND) or contrast thresholds anywhere in the display gamut are determined. First, transformation from the nonlinear RGB colour space of the display medium into the perceptual space is used. A relationship which yields the contrast threshold of a stimulus given the values of the colour values of the three components is also required. Fortunately, determining the chrominance contrast can be avoided and the luminance contrast is concentrated on, because changes in the chrominance will not make as large a contribution as changes in luminance. The analysis can therefore be restricted to determining the contribution that each of the three colour components has on the luminance contrast.

To determine the transformation from the display gamut to the perceptual gamut the relationship of the luminance (L) to the display phosphor chromaticity is specified by :

$$L = 0.272\,R + 0.562\,G + 0.103\,B \qquad (1)$$

This relationship assumes that the tristimulous R,G,B values give linear responses. In reality they are nonlinear since much lower resolution is required for low image intensity values. Three more variables Rg, Gg, Bg are defined therefore to denote the gun voltages expressed in pixel intensity units at each gun for the R,G,B components. For the above luminance relationship to hold each of the R,G,B components must range from 0 to 1 but the Rg, Gg, Bg settings vary from 0 to 255 so that scaling is necessary. The gamma value for each of the guns is marginally different but the exact nonlinearity is not critical and good results are obtained with approximate values. Similarly small changes in primaries produce little effect. The relationship between the gun voltages and the luminance is:

$$L = 0.272\left(\frac{R_g}{256}\right)^{\gamma} + 0.562\left(\frac{G_g}{256}\right)^{\gamma} + 0.103\left(\frac{B_g}{256}\right)^{\gamma} \qquad (2)$$

From Weber's Law the contrast is proportional to the logarithm of the luminance. While various slightly different models exist, one particular model states that C = log10 L, and the contrast ratio ΔC is defined as:

$$\Delta C = \partial\left(\log_{10} L\right) = \frac{f'(L)}{L \log_e 10} \tag{3}$$

Now each component of L is an independent variable and the luminance contrast is also independent of the chrominance. Therefore substituting for the luminance the rate of change of the luminance contrast is related to the rate of change of the red gun voltage according to the following relationship (similarly for Gg and Bg).

$$\frac{\partial\left(\log_{10} L\right)}{\partial R_g} = \frac{0.272(1/256)^{\gamma}\,\gamma\, R_g^{\gamma-1}}{2.3\left[0.272\left(\frac{R_g}{256}\right)^{\gamma} + 0.562\left(\frac{G_g}{256}\right)^{\gamma} + 0.103\left(\frac{B_g}{256}\right)^{\gamma}\right]}$$

By rearranging this and substituting for the contrast ratio ΔC the required change in a given component to produce a desired contrast ratio is obtained.

$$\partial R_g = \Delta C \cdot \frac{2.3\left(1/256\right)^{\gamma}\left[0.272R_g^{\gamma} + 0.562G_g^{\gamma} + 0.103B_g^{\gamma}\right]}{0.272(1/256)^{\gamma}\,\gamma\, R_g^{\gamma-1}} \tag{4}$$

Some care must be taken with this relationship because it is based on determining the required change in any one component in order to exceed the contrast threshold. A simultaneous change in two components may exceed the contrast threshold by a large enough amount to be noticeable. To solve for this the gradient of a function is used which is defined as $\nabla f = \left\langle \frac{\partial f}{\partial R}, \frac{\partial f}{\partial G}, \frac{\partial f}{\partial B} \right\rangle$

so that $df = \frac{\partial f}{\partial R} dR + \frac{\partial f}{\partial G} dG + \frac{\partial f}{\partial B} dB$, hence:

$$\Delta C = \frac{\partial L}{\partial R} dR + \frac{\partial L}{\partial G} dG + \frac{\partial L}{\partial B} dB$$

Rearranging each of the components:

$$dR = \left(\Delta C - \frac{\partial L}{\partial G} dG - \frac{\partial L}{\partial B} dB \right) / \frac{\partial L}{\partial R}$$

$$dG = \left(\Delta C - \frac{\partial L}{\partial R} dR - \frac{\partial L}{\partial B} dB \right) / \frac{\partial L}{\partial G}$$

$$dB = \left(\Delta C - \frac{\partial L}{\partial G} dG - \frac{\partial L}{\partial R} dR \right) / \frac{\partial L}{\partial B}$$

These three equations have to be solved simultaneously since the partial differentials are all interdependent.

Another aspect that must be also considered under normal viewing conditions is the stray and reflected ambient luminance that enters the eye. This tends to desaturate the colour and decrease the overall contrast. This problem manifests itself most noticeably at low screen intensity (dark colours) and under high ambient light intensities. The perceived luminance is specified in terms of both the emitted luminance from the display plus the reflected ambient light. According to Weber's Law the contrast ratio ΔC is defined in terms of the luminance of the surround (Ls) and the luminance of the stimulus (Lo) in the form:

$$\frac{|\ Ls\ -\ Lo\ |}{Lo} = \frac{\Delta L}{L} \approx \Delta C \qquad (5)$$

Where $\Delta L = |\ Ls - Lo\ |$ and $L = Lo$.

Adding the ambient luminance factor *La* :

$$\frac{|(Ls + La)\ -\ (Lo + La)|}{Lo + La} = \frac{\Delta L}{L + La} \approx \Delta C \qquad (6)$$

Which becomes :

$$\frac{\Delta L}{L} \approx \Delta C \left(1 + \frac{La}{L} \right) \qquad (7)$$

The ambient luminance modifies the contrast ratio by this significant factor. This augmented term can be substituted for the contrast ratio into the relationship given in equation 4 to account for the effects of ambient luminance on the required gun voltage to effect a desired contrast ratio.

$$\partial R_g = \Delta C\left(1+\frac{La}{L}\right)\frac{2.3\left[0.272R_g^{\gamma}+0.562G_g^{\gamma}+0.103B_g^{\gamma}\right]}{0.272\gamma\ R_g^{\gamma-1}}$$

Where L is defined in terms of equation (2). From Weber's Law the contrast ratio (ΔC) for detection (threshold) is a constant value of 0.02 for a very wide range of conditions. For the red component the amount of change required to exceed the JND threshold anywhere in the gamut becomes (similarly for the others) :

$$\partial R_g = \frac{0.046\left[0.272R_g^{\gamma}+0.562G_g^{\gamma}+0.103B_g^{\gamma}+256^{\gamma}La\right]}{0.272\gamma\ R_g^{\gamma-1}}$$

The gamma value is typically around 2.2. The maximum brightness for colour display CRTs is around 80-100 cd/m² and a very conservative amount for the reflected light is around 4 cd/m² (50 Lux). The reflected light should be normalised with respect to the display brightness as a percentage. These equations provide the required quantisation step size is so that the quantisation error remains below the perceivable threshold. A quantiser operating in the display RGB space can now be directly formulated which is approximately perceptually uniform and visually lossless. The equations are however slightly complex for real-time implementation. This problem can be overcome by estimating a zeroth order piece wise linear approximation to these relationships. This was done by plotting the relationships and then manually fitting a piece wise linear approximation to the curves. These approximations require only simple functions, such as provided by look up tables, to perform perceptually uniform and visually lossless colour quantisation.

The gun voltages are specified as an unsigned 8 bit number. Since the quantisation step sizes are all powers of two the quantisation process of the perceptual colour reduction process only involves applying a bit mask to each colour value where the mask is determined based on the value of the three components and the neighbouring changes in component values. A real-time

implementation for the perceptual colour reduction stage is shown in Figure 4, and only requires a look up table and memory which specifies the quantiser step size anywhere in the colour space.

The colour management component 16 of the input colour processing, manages all colour changes in the video. The colour data produced by the prior stages in the colour processing results in a set of displayed colours. This set is dynamic given that is creation is an adaptive process. Selecting an appropriate scheme to manage the adaptation of the colour map is important. Three distinct possibilities exist for the colour map, it may be static or segmented and partially static or fully dynamic. With a fixed or static colour map the local image quality will be reduced, but high correlation is preserved from frame to frame, leading to high compression gains. In order to maintain high quality images for video where scene changes may be frequent the colour map must be able to adapt instantaneously. Selecting a new colour map for each frame is optimal in this sense. However this has a high bandwidth requirement since not only must the entire colour map be replaced every frame but also a large component of the pixels in the image would need to be remapped each time. This remapping also introduces the problem of colour map flashing. A compromise is to permit limited colour variation between successive frames. This can be achieved by partitioning a colour map into static and dynamic sections or by limiting the number of colours that are allowed to vary per frame. In the first case only the entries in the dynamic section of the table can be modified which ensure that certain predefined colours will always be available. In the other scheme there are no reserved colours and any may be modified. While this approach helps preserve some data correlation the colour map may not be able to adapt quickly enough in some cases to eliminate image quality degradation. Existing approaches are willing to trade image quality to ensure that the image correlations are not destroyed. For any of the dynamic schemes colour map methods, synchronisation is required to preserve temporal correlations. This synchronisation process has three components.

1. Ensuring that colours carried over from each frame into the next are mapped to the same indexes over time. This involves resorting each new colour map in relation to the current one.

2. A replacement scheme is used for updating the changed colour map. To reduce the amount of colour flashing the most appropriate scheme is to replace the obsolete colour with the most similar new replacement colour.

3. Finally, all existing references in the image to any colour which is no longer supported are replaced by references to currently supported colours.

The colour difference management component 16 is also responsible for calculating the perceived colour difference at each pixel between the current and preceding frame. This information is then passed onto the spatial/temporal coding component 18 in the video encoding process. This information consists of which regions in the frame are fully transparent, which have changed more than a given amount determined by an adaptive threshold, and what needs to be replenished. To ensure that prediction errors do not accumulate and degrade the image quality a loop filter is used. This forces the frame replenishment data to be determined from the present frame and the accumulated previous transmitted data (the current state of the decoded image), not from the present and previous frames. This process is shown in Figure 7.

Following the colour map management 16 the next component 18 of the video encoder takes the now indexed colour frames and compresses them using a simultaneous intra/inter method. This uses a tree splitting method to recursively partition each frame into smaller polygons according to a splitting criteria. A quad tree split method used, as is shown in Figure 5. This attempts to represent the image by a uniform block, the value of which is equal to the global mean value of the image. If at some locations of the image the difference between this representative value and the real value exceeds some tolerance threshold then the block is recursively subdivided uniformly, into two or four subregions and a new mean calculated for each subregion. For lossless image encoding there is no tolerance threshold. Potentially every pixel in the image must be explicitly encoded and data expansion may occur instead of compression. The tree structures are composed of nodes and pointers, each node represents a region and contains pointers to any child nodes, representing subregions which may

exist. There are two types of nodes, leaf and non leaf nodes. Leaf nodes are those which are not further decomposed and as such have no children, instead containing a representative value for the implied region. Non-leaf nodes do not contain a representative value since these consists of further subregions and as such contain pointers to the respective child nodes, these can also be referred to as parent nodes.

In this video combined inter/intra frame encoder, the tree representation is used for both motion compensated coding and image value coding. The process starts by considering a region consisting of the entire frame and a decision is made as to whether it is to be split. To do this a search is first made in the neighbour hood of the region to determine if the region has been displaced from the previous frame, and this is often determined with subpixel accuracy. This can be performed using an exhaustive search or one of a number of faster search techniques, such as the 2D logarithmic, three step and simplified conjugate direction search, as shown in Figure 6. The aim of this search is to find the displacement vector for the region often called the motion vector. This is found by locating the displacement for a region where the number of pixels that are different in the previous frame compared to the current frame region are the least if the region is not transparent. Once the motion vector is found then the region is motion compensated by predicting the value of the pixels in the region from their original location in the previous frame according to the motion vector. The motion vector may be zero if the vector giving the least difference corresponds to no displacement. If the number of pixels in the region that is different from the previous frame is greater than a given adaptive threshold then the block needs to be split further. Otherwise either the motion vector (in 8 bits) or the region colour value (in 8 bits) if there is no displacement is encoded for the region. If the region is to be split then this entire process is repeated with each new subregions

The rationale behind the 2D logarithmic search is to follow the direction of minimum distortion. At each step of the algorithm five locations are checked one at the centre of the block and the other midpoints between the centre and the four boundaries of the search area. If the minimum distortion is at the centre location then the distance between the search points is decreased, otherwise the point of

least distortion becomes the new centre of the search area. A new step is initiated and the search continues until the search area becomes a square of 3x3 pixels.

In the three step search eight evenly spaced locations are initially tested around the centre pixel and the centre is translated to the location which gives the minimum prediction error. In the second step another eight locations are tested around the new centre but this time the spacing between them is reduced. This procedure is repeated a total of three times, with the location giving rise to the least prediction error of the last step being selected as the displacement vector.

The simplified conjugate direction search performs the searching first in the horizontal and then vertical directions. The object of the search is to locate the direction which gives the minimum prediction error. Initially it looks on either side of the centre point and having located the direction which minimises the prediction error it will then analyse the next point in that same direction. This continues until the next point in that direction no longer decreases the prediction error. Once this occurs and the current point is located between two points giving higher error values then the search in the vertical direction begins and is executed in the same fashion as the horizontal search.

The actual video frame data is encoded using a preordered tree traversal method. There are actually three types of leaves in the tree, transparent leaves, motion compensation leaves and region colour leaves. The transparent leaves indicate that the region indicated by the leaf is completely transparent. The motion compensation leaves contain motion vectors and the colour leaves contain the region colour. The encoder starts at the top of the tree and for each node stores a single ONE bit if the node is a leaf, in this case if the region is a transparent region then another single ZERO bit is stored otherwise another single ONE bit is stored. This is followed by either another single ONE bit and the motion vector in 8 bits or by a single ZERO bit and the colour value of the region in 8 bits. If the node was a parent node then a single ZERO bit is stored and each of the child nodes are then stored. Entropy coding is then applied to the stored tree data to further compress it.

For improved image quality, additional colour information can be transmitted. This is performed by constructing a smoothed image from the encoded data before it is stored for transmission. This is performed using cubic interpolation and the resulting image is subtracted from the original frame data. The psychophysical masking effect is also determined for the original image and used to determine a local threshold. This difference image is scanned for all locations where the difference exceeds the masking threshold, and additional 24 bit colour difference information is stored for transmission. Since this data will be non-uniformly distributed as a spare matrix the data is stored using a pseudo-random, adaptive technique based on double prediction.

This is a form of serpentine raster scanning where the direction of the scan is predicted and only an error signal is encoded which may temporarily inhibit the normal reversal of the scan direction every time the frame boundary is reached. The operation, as shown in Figure 8, is as follows. A scan will commence up or down a column and when the last pixel to be encoded has been visited, note is taken of its vertical position. If its position is past the middle of the scan then the next scan will be in the opposite direction, otherwise it will be in the same direction as the preceding one. If in the process of performing the next scan the first pixel encountered is in the expected direction relative to the prediction based on the last pixel in the previous column then a positive polarity is assigned to the start of the scan, if the pixel is not in the expected position then a negative polarity is assigned to the transition. With this scheme only the polarities of each transition are encoded which represent the error signal for the scan direction predictor. Most rows or columns are only partially visited, and many not at all. The expected scan direction is based on the position of the last encoded pixel in the previous line. The scan direction is the actual direction to be travelled in order to visit the first pel in the next scan and the polarity is the error from the predicted or expected scan direction. In this procedure only the polarity of the prediction error is transmitted and only if the relative distance of the next data point in the horizontal direction is non zero. This is only one bit of data and otherwise no scan direction information is encoded at all. The scheme functions in the following manner :

```
H = height of raster
last data point = 0
IF at start of new column
        IF last data point > H / 2
                THEN expected scan direction = -ve
                ELSE expected scan direction = +ve
        IF predicted scan direction = +ve AND current point > last point
                THEN Polarity = +ve;
                ELSE Polarity = -ve;
        IF predicted scan direction = -ve AND current point > last point
                THEN Polarity = -ve;
                ELSE Polarity = +ve;
END
```

The object scene control data input to the component 14 permits each object to be associated with one visual data stream, one audio data stream and one of any other data streams. It also permits various rendering, and presentation parameters for each object to be dynamically modified from time to time throughout the scene. These include the amount of object transparency, object scale, object volume, object position in 3D space, and object orientation (rotation) in 3D space.

The compressed video and audio data is now transmitted or stored for later transmission as a series of data packets. There are a plurality of different packet types. Each packet consists of a common base header and the payload. The base header identifies the packet type, the total size of the packet including payload, what object it relates to and a sequence identifier. The following types of packets are currently defined SYSCTRL, VIDEO, AUDIO, TEXT, VECTOR, VIDCTRL, AUDCTRL, TXTCTRL, VECCTRL, OBJCTRL. There are two main types of packets, control and data packets. The control packets (CTRL) are used to set or change the parameters of the system for handling the objects identified. The data packets contain the compressed information required to generate the video

objects. Except for the SYSCTRL packet the data packets appear first, whereas the other packet types may appear in any order and as frequently as required.

The SYSCTRL packet is used to define general scene information. This is always the first packet in a data stream, and defines the video identifier, the width and height of the scene space (up to 65536x65536 pixels), the total number of audiovisual objects in the scene (up to 256) and the background colour of the scene window.

The VIDCTRL packet is used to define the parameters for the video component of an object in the scene. This defines the compression class used for the data permitting a variety of different coding methods to be used. It also defines the frame play out rate and the width and height of the video frames for the identified video object.

The AUDCTRL packet is used to define the parameters for the audio component of an object in the scene. This defines the type of sound used for the data, which includes MONO, DUAL, STEREO, and JOINT STEREO. It also defines the audio sample rate and the number of samples per frame/packet for the identified audio object.

The OBJCTRL packet is used to define the parameters for the rendering and other properties of scene objects. This packet only includes one field which is a bit mask specifying what properties are defined in the corresponding data packet. There are currently 5 defined properties, object transparency, scale, volume, 3D position, and 3D rotation, and additional properties may be added.

In the decoder the input buffer 30 stores all incoming data until a full packet has been received. This is then passed to the input data switch 32 that analyses the packet and redirects the contained data to the appropriate processing module according to the packet type. The colour management module 36 receives information from video packets pertaining to colour sets. The video decoding module 38 receives and processes the rest of the information in the video packets. The audio decoding module 42 receives and processes all the audio

packet data while the object management module 40 receives the object control data.

Both the video and audio decoding modules in the decoder independently decompress any data sent to them and perform a preliminary rendering of it into a temporary buffer. The final scene rendering is then performed onto the display by transforming the video and audio components of each object according to the information conveyed within the object control packets. The video object display 44 is responsible for controlling the play back rate of the video and audio.

Another component of the object oriented video system is means for encrypting/decrypting the video stream for security of content. One preferred embodiment uses a combination of a mono or polyalphabetic substitution cipher in conjunction with a transposition cipher. The key to perform the decryption is separately and securely delivered to the end user, perhaps by encoding it using the RSA public key system.

An additional security measure is including a universally unique brand/identifier in an encoded video stream. This could take at least four principal forms:

a. In a videoconferencing application a single unique identifier is applied to all instances of the encoded video streams

b. In broadcast video-on-demand (VOD) with multiple video objects in each video data stream, each separate video object has a unique identifier for the particular video stream.

c. A wireless ultrathin client system has a unique identifier which identifies the encoder type as used for wireless ultrathin system server encoding as well as identifying a unique instance of this software encoder.

d. A wireless ultrathin client system has a unique identifier which uniquely identifies the client decoder instance in order to match the Internet based user profile to determine the associated client user.

The ability to uniquely identify a video object and data stream is particularly advantageous. In videoconference applications there would be no real need to monitor/log the teleconference video data streams except where advertising

content may occur (which would be uniquely identified as per the VOD). The client side decoder software would log viewed decoded video streams (identifier, duration). Either in real time or at subsequent synchronisation this data would be transferred to an Internet based server. This information would be used in the generation of marketing revenue streams as well as market research/statistics in conjunction with client personal profiles.

In VOD the decoder can be restricted to decode broadcast streams or video only when enabled though holding the right key. Enabling can be performed, either in real time if connected to the Internet or at a previous synchronisation of the device, when accessing an Internet authentication/access/billing service provider which provides means for enabling the decoder through authorised payments or possibly to pay for previously viewed video streams. Similarly to the advertising video streams in the video conferencing, the decoder would log VOD related encoded video streams along with duration of viewing. This information would be transferred back to the Internet server for market research/feedback, and payment purposes.

In the wireless ultrathin client (NetPC) application there would be real time encoding, transmission and decoding of video streams from Internet or otherwise based computer servers which would add a unique identifier to the encoded video streams. The client side decoder would only be capable of decoding the video stream once enabled. Enabling of the client side decoder could occur along the lines of the authorised payments in the VOD application or more likely through a secure encrytion key process which may enable certain levels of access to wireless NetPC encoded video streams. The computer server encoding software may include multiple levels of access. In the broadest form, ubiquitous wireless Internet connection would include mechanisms for monitoring client connections through decoder validation fed back from the client decoder software to the compute servers. Computer servers can therefore monitor client usage of server application processes and charge accordingly or monitor streamed advertising to end clients.

As indicated there are a variety of different application scenarios for the video system. For example, in the context of delivering object oriented video content to Internet connected users there are three possible modes. The first mode is to deploy the object oriented video coder/decoder as registered application for world wide web browsers. In this mode users may select video content to download by selecting a hyperlink pointing to a small parameter file on HTML page. The client WWW browser then invokes a special video client program which reads the parameters in the downloaded file and uses these to open a network connection to the relevant video server program and request the appropriate video data stream to download. In the second mode users may operate a remote HTML browser using the ultra thin client system to be described below. The final mode requires the use of new markup and scripting language, hereinafter referred to as IVML (interactive video markup language), that has been designed to coexist on the same level with HTML. With IVML content creators are able to define interactive, object oriented, audio-visual information spaces, in much the same way that HTML content for the WWW is currently created.

IVML is similar in some respects to HTML but is specifically designed to be used with object oriented multimedia spatio-temporal spaces such as audio/video. It may be used to define the logical and layout structure of these spaces, including hierarchies, it may also be used to define linking, addressing and also metadata. This is achieved by permitting five basic types of markup tags to provide descriptive and referential information etc. These are system tags, structural definition tags, presentation formatting, and links and content. IVML is not case sensitive and each tag comes in two forms, opening and closing which are used to enclose the parts of the text being annotated. For example:

<TAG> some text in here </TAG>

Comments may be included by using a special tag:

<!-- Put any text in here -->

Other special tags to control the web server include:

<!--#include file="FileName" -->

The following system tags are currently defined:

<IVML>	marks the start and end of the file
<HEAD>	descriptive info such as header
<TITLE>	for browser, must be in header
<BODY>	your page content
<BASE HREF="URL">	where am I, [no end tag]
<BASE TARGET="name">	my file name
<META>	meta information may be anywhere
<SCRIPT>	For Scripting animation

A minimal IVML page (or node or file) should consist of:

```
<IVML>
<HEAD>
    <TITLE> Que? </TITLE>
    <BASE HREF="http://URL">
</HEAD>
<BODY>
    <!-- Put your page content here -->
</BODY>
</IVML>
```

Referential tags for creating links and identifying content include <A HREF> and <A NAME>. The HREF tag is for defining links to something. The link URL can point to another document, a particular target in another document or a target within itself. Targets can be any structural definition tags or a generic naming tag:

<A HREF= "URL"> text </A> - in other document

<A HREF= "#name"> text </A> - in same document

The targets or anchor points are defined using:

<A NAME = "what_dya_call_it"> text </A>

IVML Video and audio data can be displayed by specifying the URL of the video or audio file:

<VID SRC = "URL"> and <AUD SRC = "URL">

Structural definition of audio-visual spaces uses structural tags and include the following:

<A SCENE>	Defines video scenes
<A OBJECT>	Defines object properties, composition
<A VIDOBJ>	Defines video object data
<A AUDOBJ >	Defines audio object data
<A TXTOBJ>	Defines text object data
<A VECOBJ>	Defines vector object data
<A FRAME>	Defines video frame
<A PATH>	Defines animation path data

Layout definition of audio-visual objects uses layout tags to define the spatio-temporal placement of objects within any given scene and include the following:

<SCALE>	Scale of visual object
<VOLUME>	Volume of audio data
<ROTATION	Orientation of object in 3D space
<POSITION>	Position of object in 3D space
<TRANSPARENT>	Transparency of visual objects
<TIME>	Start time of object in scene
<PATH>	Animation path from start to end time
<HREF>	Link tag

Presentation definition of audio-visual objects uses presentation tags to define the presentation of objects and include the following:

<SCENESIZE>	Scene spatial size
<SCENECOLR>	Scene background colour
<SCENEOBJ>	Scene background object
<TRANSITION>	Scene Transition effects
<VIDRATE>	Video Frame rate
<VIDSIZE>	Size of video frame
<VIDEFFECT>	Scene change effects
<AUDRATE>	Audio sample rate
<AUDBPS>	Audio sample size in bits
<TXTFONT>	Text Font type to use
<TXTSIZE>	Text font size to use
<TXTSTYLE>	Text style (bold, underline, italic)
<TXTFORM>	Paragraph style (centred, left, right)
<TXTSTYLE>	Text style (bold, underline, italic)
<TXTCOLR>	Text colour to use

Scene transition effects includes the following types:

<FADEIN>	Fade from black to image
<FADEOUT>	Fade from image to black
<DISOLVE>	Cross disolve from image A,B
<WIPE>	Scene Transition effects

An IVML file will generally have one or more scenes and one script. Each scene is defined to have a determined spatial size, a default background colour and an optional background object in the following manner:

<A SCENE = "sceneone">

< SCENESIZE SX = "320", SY="240">

< SCENECOLR ="#RRGGBB" >

<A VIDOBJ SRC = "URL">

<A AUDOBJ SRC = "URL">

<A TXTOBJ > this is some text string

</ A>

Alternatively the background object may have been defined previously an then just declared in the scene:

<A OBJECT = "backgrnd">

<A VIDOBJ SRC = "URL">

<A AUDOBJ SRC = "URL">

<A TXTOBJ > this is some text string

<SCALE = "2'>

<ROTATION = "90">

<POSITION= XPOS ="50" YPOS="100">

<HREF = "URL">

</A>

<A SCENE>

< SCENESIZE SX = "320", SY="240">

< SCENECOLR ="#RRGGBB" >

<SCENEOBJ = "backgrnd">

</ A>

Scenes can contain any number of foreground objects:

<A SCENE>

< SCENESIZE SX = "320", SY="240">

< SCENECOLR ="#RRGGBB" >

< OBJECT = "foregnd_object1", PATH ="somepath">

<OBJECT = "foregnd_object2", PATH ="someotherpath">

<OBJECT = "foregnd_object3", PATH ="anypath">

</ A>

Paths are defined for each animated object in a scene:

<A PATH = somepath>

< TIME START="0", END="100">

< POSITION TIME=START, XPOS="0", YPOS="100">

< POSITION TIME=END, XPOS="0", YPOS="100">

<INTERPOLATION= LINEAR>

</ A>

The script specifies the scene playout structure and presentation information regarding scene transitions:

<SCRIPT>

<TRANS = "sceneone", FADEIN>

<PLAY = "sceneone">

<TRANS = "sceneone", "scenetwo", DISOLVE>

<PLAY = "scenetwo">

<TRANS= "scenetwo", FADEOUT>

</SCRIPT>

Using IVML content creators can textually create animation scripts for object oriented video. More advanced editing software programs may also be created to permit content creators to edit object oriented video in a graphical manner to automatically create the IVML file. The structure and meta data fields permit flexible access to and browsing of the object oriented video. The hyperlink references within the file permit objects to be clicked on that invoke defined actions. After creation of an IVML file the IVML server program may then processes the file to create the composite video stream that is delivered to a users who requests the IVML document from the server.

By using an ultra thin client as a device for controlling a first computer of any kind from any other kind of personal mobile computing device, enables the creation of virtual computing networks. In this new application, the user's computing device performs no data processing but only serves as a user interface into the virtual computing network. All the data processing is performed by compute servers

located in the network. At most the terminal is limited to decoding all output and encoding all input data, including the actual user interface display. Architecturally the incoming and outgoing data streams are totally independent within the user terminal. There is no interplay within the terminal itself between these data streams. Control over the output or displayed data is performed at the compute server where the input is data is processed. Accordingly the graphical user interface (GUI) decomposes into two separate data streams; the input and the output display component which is a video. The input stream is a command sequence that may be combination of ASCII characters together with mouse or pen events. To a large extent decoding and rendering the display data comprises the main function of such a terminal and complex GUI displays can be rendered.

Figure 9 shows an ultra thin client system operating in a wireless LAN environment. This system may have a range of 300 meters indoors to up to 1 km outdoors. The ultrathin client is a personal digital assistant or palmtop computer with a wireless network card and antenna to receive signals. The wireless network card interfaces to the personal digital assistant through a PCMCIA slot or a compact flash port. The compute server may be any computer running a GUI that is connected to the internet or a local area network with wireless LAN capability. The computer server uses the video encoder to convert the GUI display and any audio to compressed video using the process described previously and transmits it to the ultra thin client. The GUI display may be captured using a GUI screen reading means which is a standard function in many operating systems such as CopyScreenToDIB() in Microsoft Windows NT. The ultra thin client receives the compressed video and renders it appropriately to the user display using the video decoder. Any user control data is transmitted back to compute server, where it is interpreted and then used to control the compute server. This control include executing new programs, terminating programs, performing operating system functions, and any other functions associated with the running program(s). This control may be effected through various means, in the case of MS Windows NT the Hooks/JournalPlaybackFunc() can be used.

For longer range applications the WAN system of Figure 10 is preferred. In this case the compute server, is directly connected to a standard Telephone interface

for transmitting the signals across a CDMA or GSM cellular phone network. The ultra thin client in this case comprises a personal digital assistant with a modem connected to a phone. All other aspects are the same. In a variation of this system the PDA and phone are integrated into a single device. Suitable devices have recently become commercially available. In one instance of this ultra thin client system the mobile device has full access to the compute server from any location whilst within the reach of standard mobile telephony networks such as CDMA or GSM. A cabled version of this system may also be used which dispenses with the mobile phone so that the ultra thin computing device is connected directly to the standard cabled telephone network through a modem.

The compute server may also be remotely located and connected via the Internet to a local wireless transmitter/receiver as depicted in Figure 11. This ultra thin client application is especially relevant in the context of the emerging Internet based virtual computing systems.

Figure 12 shows a multiparty wireless videoconferencing system involving two or more wireless client telephony devices. In this application two or more participants may set up a number of video communication links among themselves. There is no centralised control mechanism, instead each participant may decide what links to activate in an multiparty conference. For example in a three person conference consisting of persons A,B,C, links may be formed between persons AB, BC and AC (3 links), or alternatively AB and BC but not AC (2 links). In this system each user may set up as many simultaneous links to different participants as they like, as no central network control is required and each link is separately managed. The incoming video data for each new videoconference link forms a new video object stream that is fed into the object oriented video decoder of each wireless device. In this application the object video decoder is run in a presentation mode where each video object is rendered according to layout rules, based on the number of video objects being displayed. One of the video objects is identified as currently active and this one is rendered in a larger size than the other objects. The selection of which object is currently active may be performed using either automatic means based on the video object with most acoustic energy (loudness/time) or manually by the user. Client telephony devices may be a

personal digital assistant or handheld personal computer with a wireless network card and antenna to receive and transmit signals. The wireless network card interfaces to the client telephony device through a PCMCIA slot, a compact flash port or other means. Each client telephony device may include a video camera for digital video capture and one or more microphones for audio capture. The client telephony device includes the video encoder to compress the captured video and audio signals, using the process described previously, which are then transmitted to one or more other client telephony devices. The digital video camera may only capture digital video and pass it to the client telephony device for compression and transmission or it may also compress the video itself using a VLSI hardware chip (an ASIC) and pass the coded video to the telephony device for transmission. The client telephony devices, which contain specific software, receive the compressed video and audio signals and render them appropriately to the user display and speaker outputs using the process previously described. This embodiment may also include direct video manipulation or advertising on a client telephony device, using the process of interactive object manipulation described previously, which can be reflected through the same means as above to other client telephony devices participating in the same videoconference. This embodiment may include transmission of user control data between client telephony devices such as to provide a means for remote control of other client telephony devices. Any user control data is transmitted back to the appropriate client telephony device, where it is interpreted and then used to control local video image and other software and hardware functions. As in the case of the ultra thin client system application there are various network interfaces which can be used.

Figure 13 is a block diagram of an interactive video on demand system with targeted user video advertising. In this system a news or video-on-demand (VOD) provider would unicast or multicast video data streams to individual subscribers. In one instance of the video decoder, an advertising area is allocated on the device screen at certain times. This advertising area can be changed either from pre-downloaded advertising stored on the device or more likely from real time encoded video streams - targetted specifically to the client device based on the client owner's server stored profile. For the targetted video based advertising, feedback and control mechanisms for video streams and viewing thereof are

used. The VOD provider maintains and operates a video server that stores compressed video streams. When a subscriber selects a program from the video server the provider's transmission system automatically selects what promotion/advertising data is applicable from information obtained from a subscriber profile database that includes information such as subscriber age, gender geographical location, subscription history etc. The advertising data, which is stored as a single video object is then inserted into the transmission data stream together with the requested video data and sent to the user. As a separate video object, the user can then interact with the advertising video object by adjusting its presentation/display properties. The user may also interact with the advertising video objects by clicking on the object to thereby send a message back to the video server indicating that the user wishes to activate some function associated with that advertising video object as determined by the VOD provider. This function may simply entail a request for further information from the advertiser, placing a video/phone call to the advertiser or some other form of control. In addition to advertising this function may be directly used by the VOD provider to promote additional video offerings such as other available channels, that may be advertised as small moving iconic images. In this case the user action of clicking on such an icon may be used by the provider to change the primary video data being sent to the subscriber or send additional data. Multiple video object data streams may be combined by the video object overlay means into the final composite video data stream that is transmitted to each client. Each of the separate video object streams that are combined may be retrieved over the Internet work by the video promotion selection means from different remote sources such as other video servers, or web cameras etc or compute servers. Again as in the other system applications of ultra thin clients and videoconferencing there are various preferred network interfaces can be used.

Internet service providers may provide user authentication and access control to video material, metering of content consumption and billing of usage. In this situation all users must register with the relevant authentication/access provider before usage of the system may occur. The authentication/access service creates a unique identifier for each user and that is automatically stored by the client system. All subsequent requests to video content providers by users must be

performed with the use of a valid user identifier. The content providers, as shown in Figure 14, then liaise with the billing service which provides a one-time access key to the user to enable the user to view the video provided by the content provider. The access control and or billing service provider keep a user usage profile which may then be sold or licensed to third parties for advertising/ promotional purposes. In order to implement billing and usage control a suitable encryption method must be deployed as previously described. In addition to this process for uniquely branding/identifying an encoded video is required as described previously.

Figure 15 is a block diagram of a video security/surveillance system. In addition to transmitting remote video to wireless handheld devices over short ranges using wireless LAN interfaces security devices are also able to transmit remote video over long distances using a standard telephone interface over a CDMA or GSM system. Other access network architectures can also be used. The security system can have intelligent functions such as motion detection alarms, automatic notification and dial out on alarm, recording and retrieval of video segments, select and switch between multiple camera inputs, and provide for user activation of multiple digital or analogue outputs at the remote location.

Many modifications will be apparent to those skilled in the art without departing from the spirit and scope of the present invention as hereinbefore described.

DATED this 22nd day of October, 1999

~~UTOVIA, INC~~. ActiveSky, Inc.

By its Patent Attorneys

DAVIES COLLISON CAVE

AUSTRALIAN PATENT OFFICE SEC 104

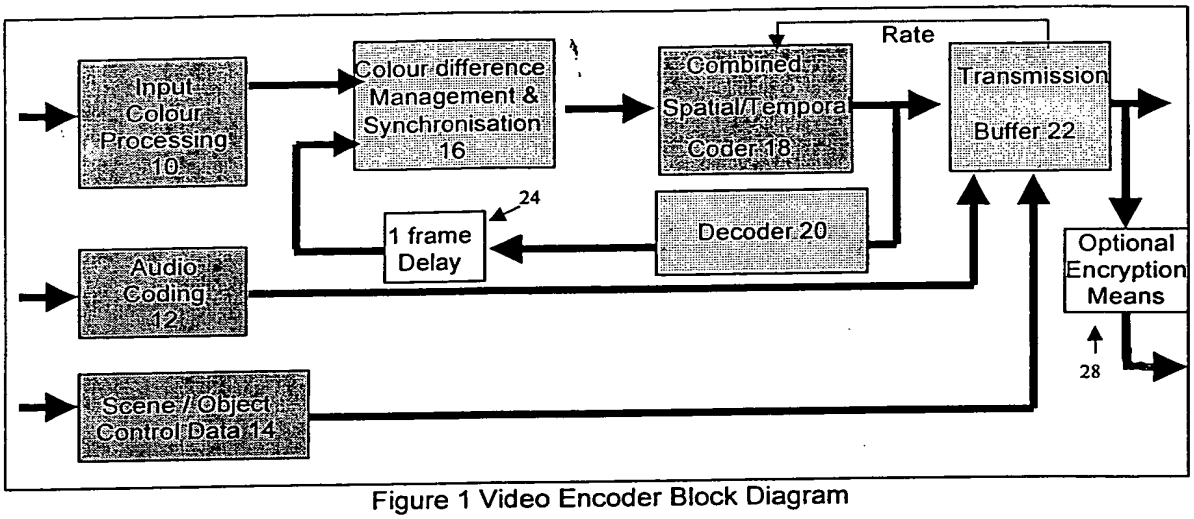

Figure 1 Video Encoder Block Diagram

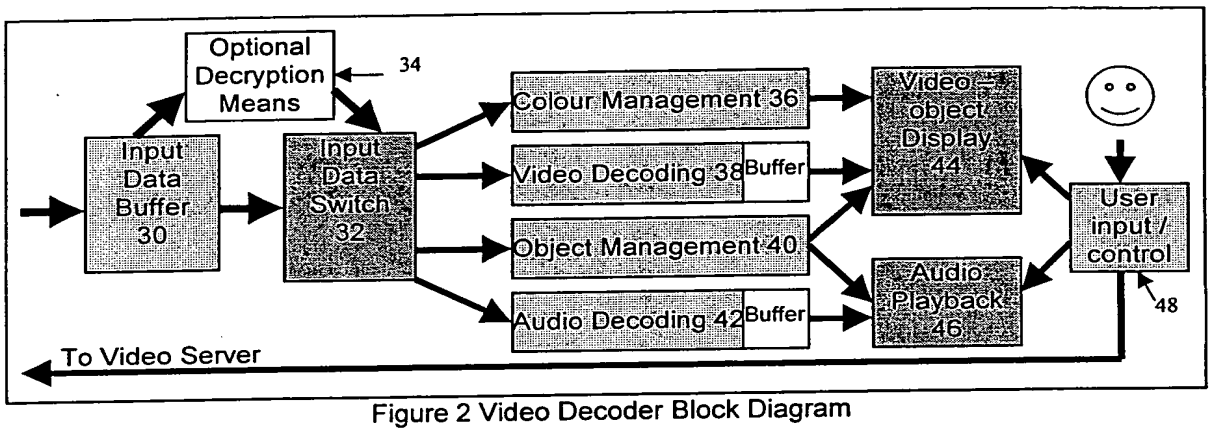

Figure 2 Video Decoder Block Diagram

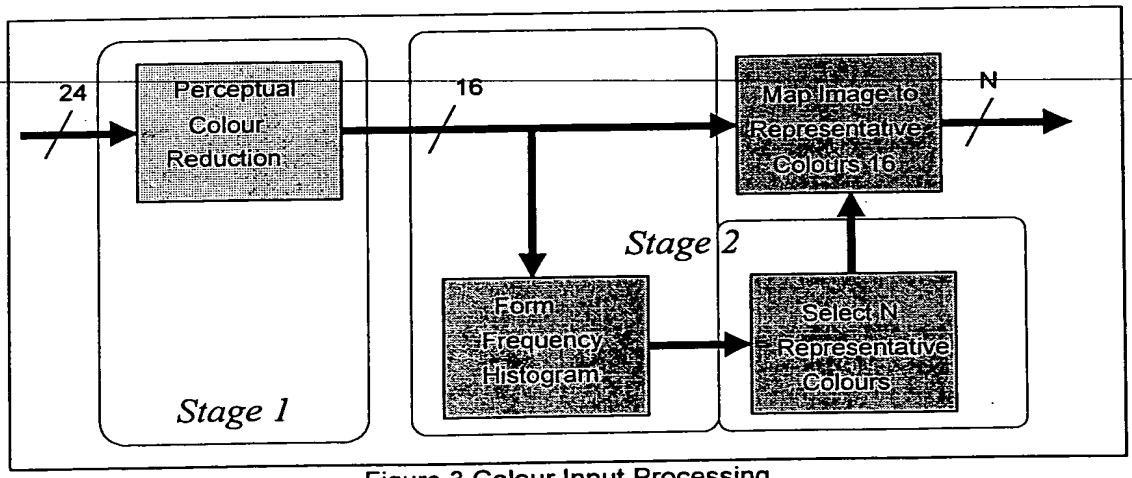

Figure 3 Colour Input Processing

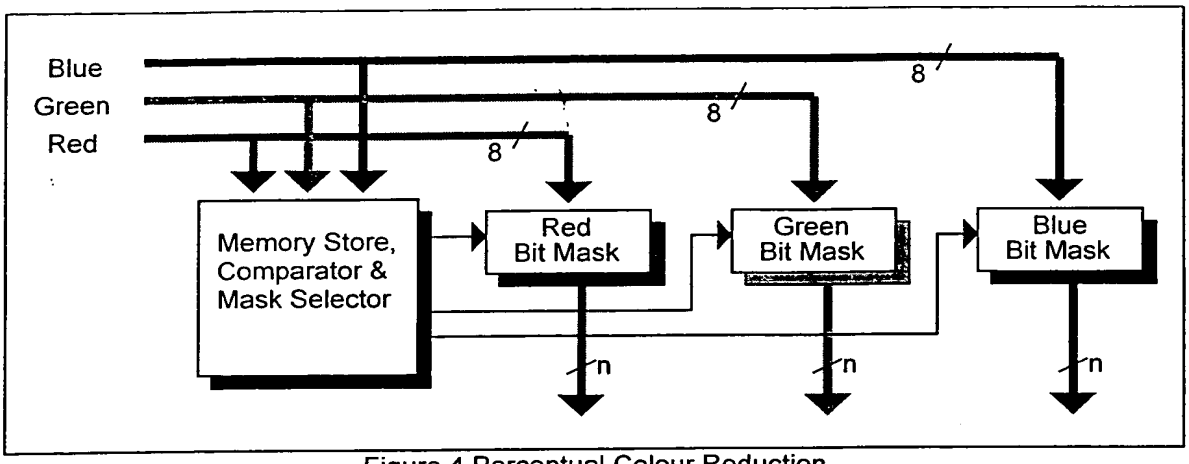

Figure 4 Perceptual Colour Reduction

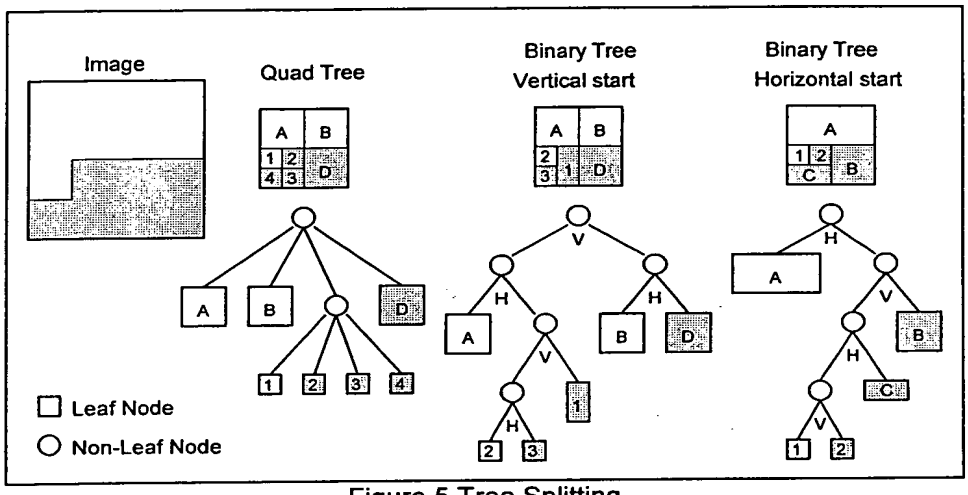

Figure 5 Tree Splitting

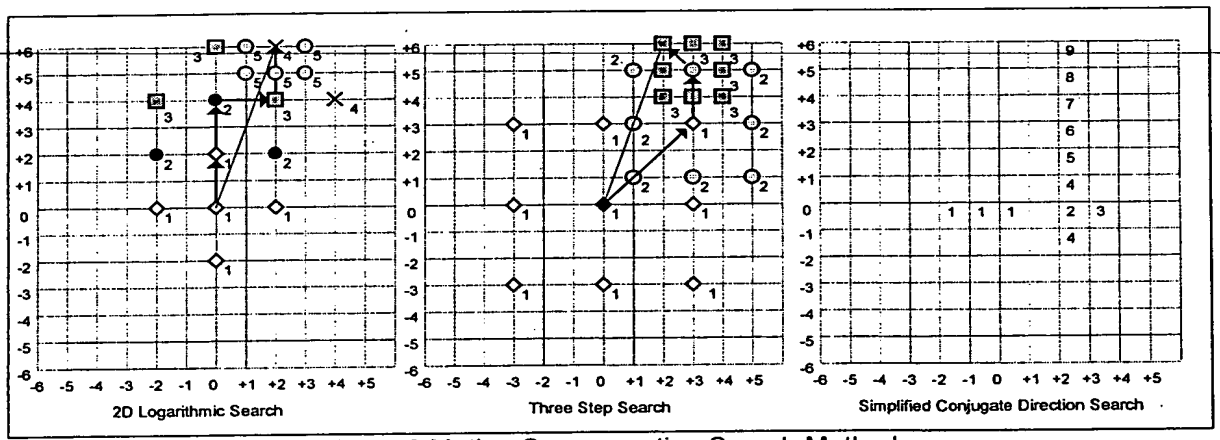

Figure 6 Motion Compensation Search Methods

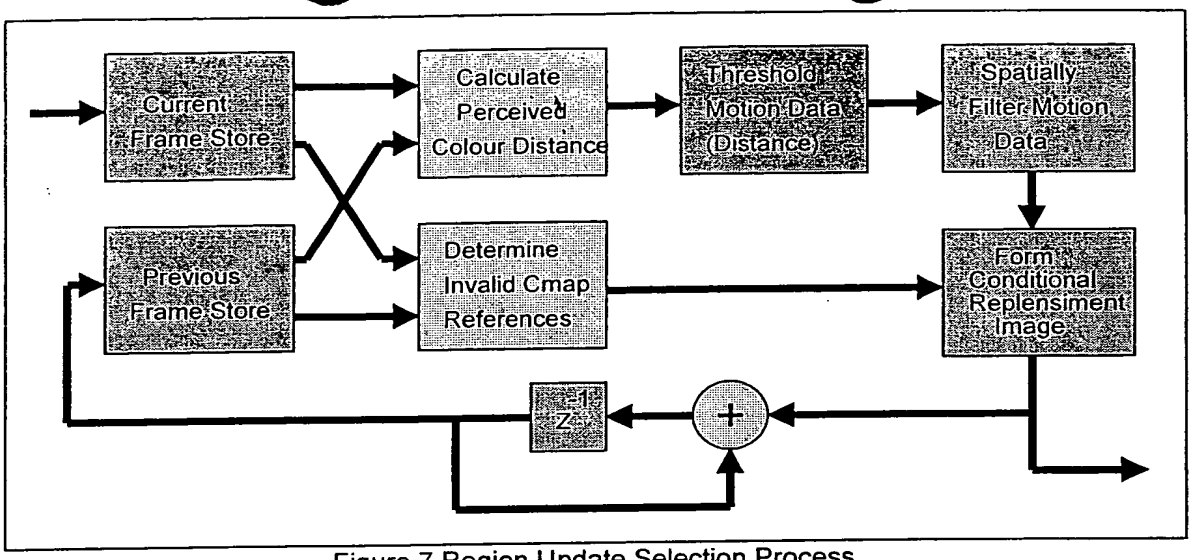

Figure 7 Region Update Selection Process

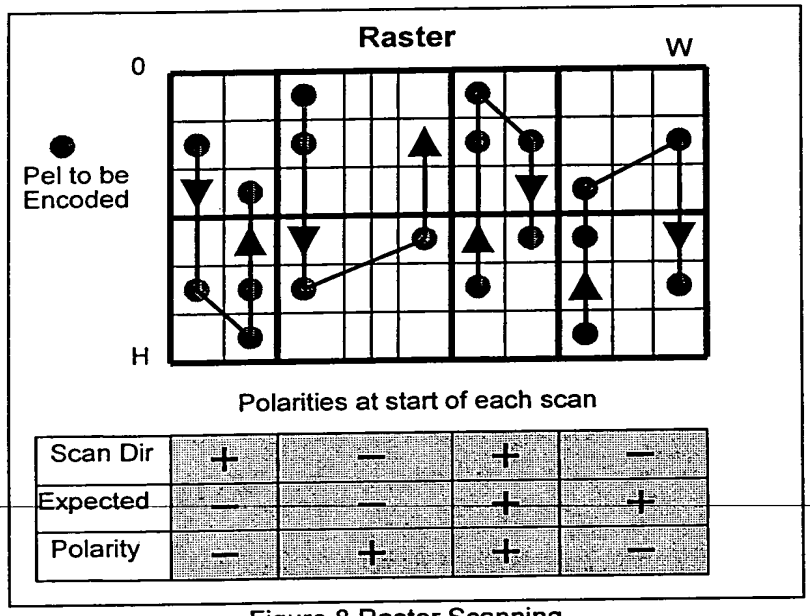

Figure 8 Raster Scanning

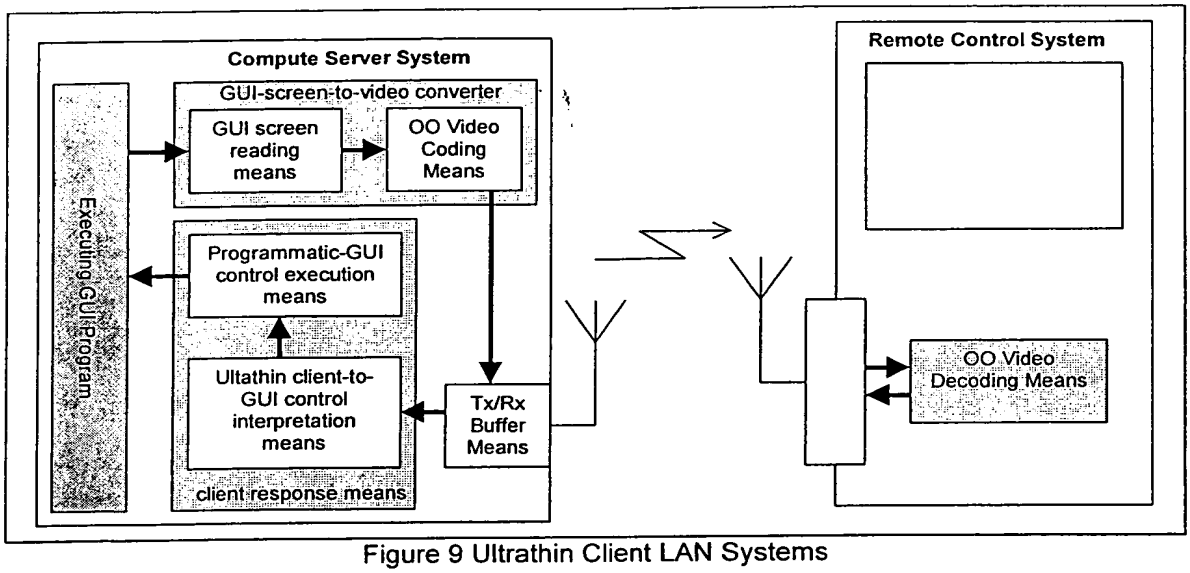

Figure 9 Ultrathin Client LAN Systems

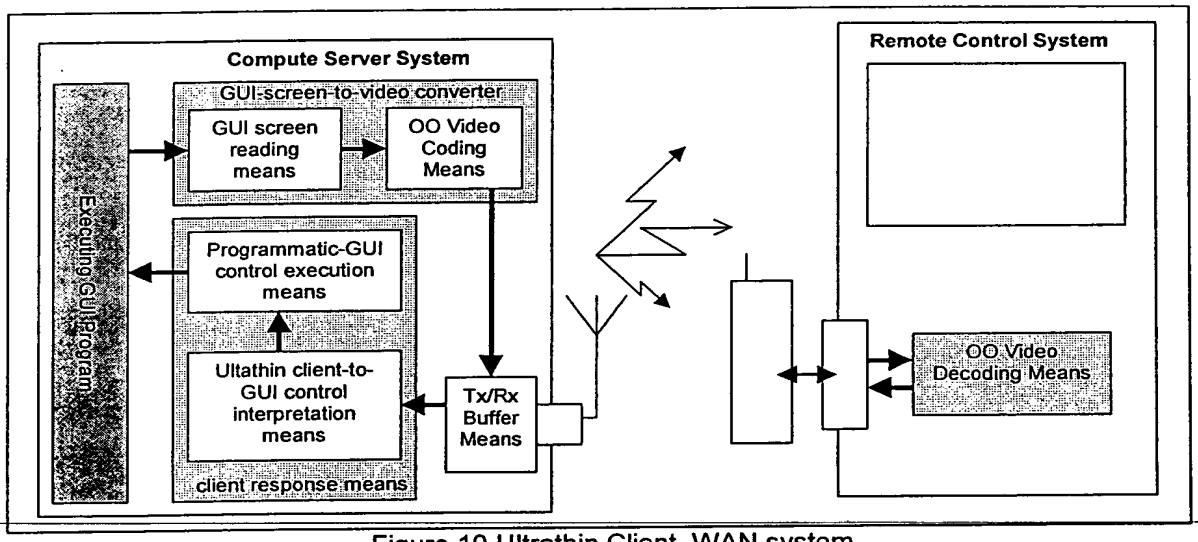

Figure 10 Ultrathin Client WAN system

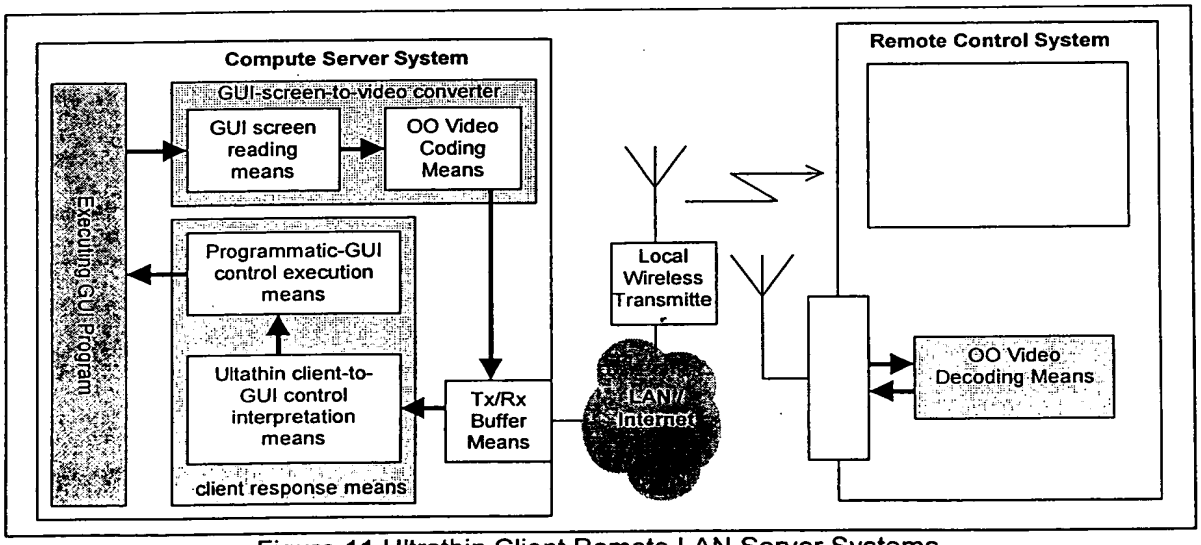

Figure 11 Ultrathin Client Remote LAN Server Systems

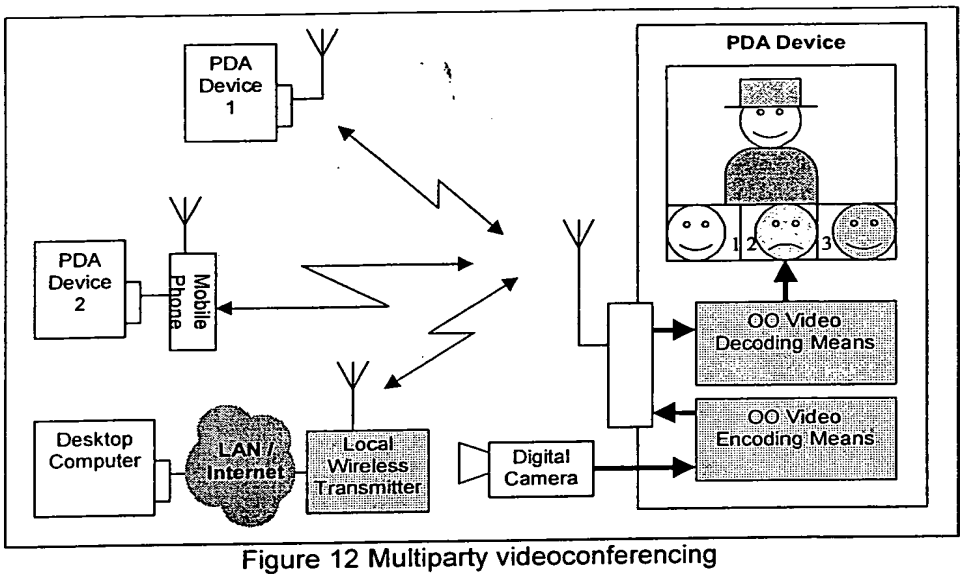

Figure 12 Multiparty videoconferencing

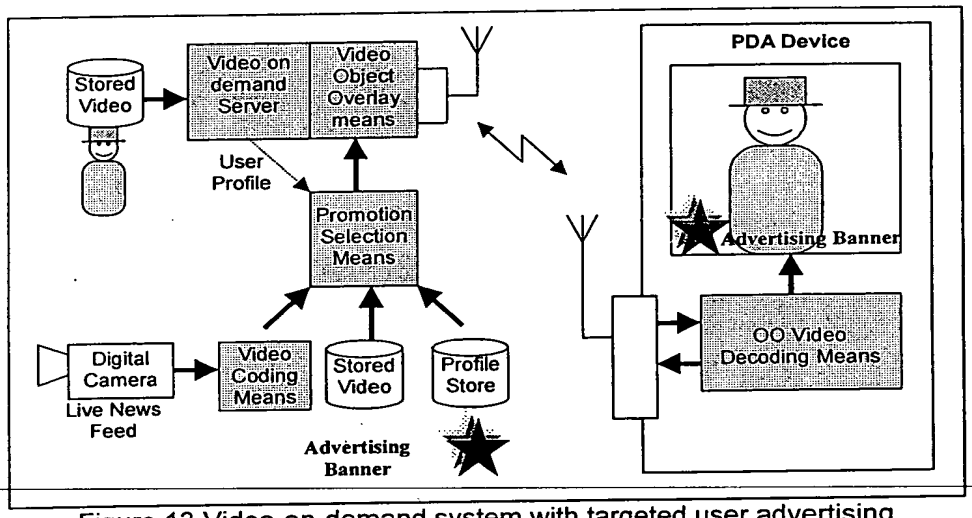

Figure 13 Video-on-demand system with targeted user advertising

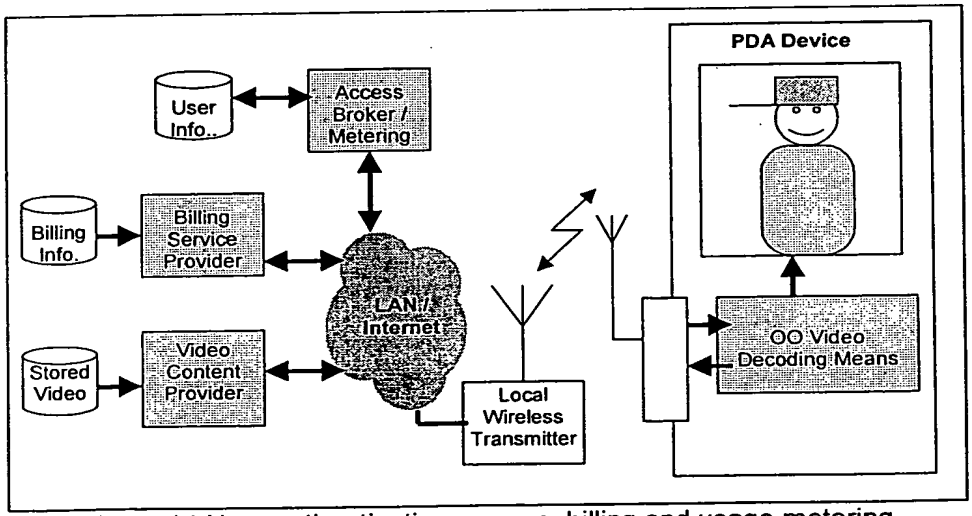

Figure 14 User authentication, access, billing and usage metering

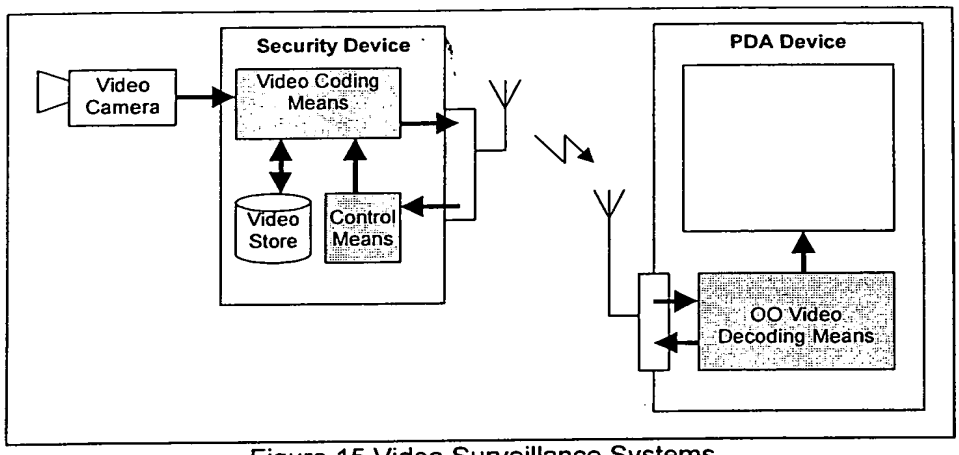

Figure 15 Video Surveillance Systems